

Draft **ES 201 218** V1.1.1 (1998-09)

ETSI Standard

**Multimedia Terminals and Applications (MTA);
Application Programming Interface (API)
for DAVIC Service Information (SI)**



Reference

DES/MTA-011074 (ba000icp.PDF)

Keywords

API, multimedia, terminal

ETSI

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet

secretariat@etsi.fr
<http://www.etsi.fr>
<http://www.etsi.org>

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 1998.
All rights reserved.

Contents

Intellectual Property Rights.....	4
Foreword	4
1 Scope.....	5
2 References.....	5
3 Definitions and abbreviations	5
3.1 Definitions	5
3.2 Abbreviations.....	6
4 Overview	6
4.1 The DAVIC application interchange format.....	6
4.2 Core set of Java APIs.....	6
5 SI tables' relationships with API classes and objects.....	7
6 Definition of getXxxx and retrieveXxxx methods.....	8
6.1 Different retrieve methods	9
6.1.1 Method 1, retrieveUpdate.....	10
6.1.2 Method 2, retrieveField and retrieveDescriptors	10
6.1.3 Method 3, retrieveSIInformation.....	10
6.2 Accessing descriptors.....	10
7 The API class diagram	10
8 The notification mechanism.....	13
9 Package org.davic.net.dvb.si.....	14
9.1 Interfaces.....	14
9.2 Class Definitions.....	22
9.3 Exception definitions	62
10 Examples.....	66
10.1 Example 1.....	66
10.2 Example 2.....	67
10.3 Example 3.....	68
History.....	69

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://www.etsi.fr/ipr> or <http://www.etsi.org/ipr>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Standard (ES) has been produced by ETSI Project Multimedia Terminals and Applications (MTA), and is now submitted for the ETSI standards Membership Approval Procedure.

1 Scope

The present document specifies an API to access the DAVIC Service Information (SI). It is applicable only to Set Top Units (STUs) in the framework of DAVIC 1.3 specification.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

- [1] DAVIC 1.3 specification.
- [2] EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".
- [3] ETR 211: "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)".
- [4] ISO/IEC 13818-1: "Information technology – Generic coding of moving pictures and associated audio information: Systems".
- [5] ISO/IEC IS 13818-6 (1996): "Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extension for Digital Storage Media Command and Control".
- [6] ISO/IEC 13522-5: "Information technology - Coding of multimedia and hypermedia information - Part 5: Support for base-level interactive applications".
- [7] ISO/IEC CD 13522-6 (1996): "Information technology - Coding of Multimedia and Hypermedia Information - Part 6: Support for enhanced interactive applications".
- [8] ETS 300 777-1: "Terminal Equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 1: Coding of multimedia and hypermedia information for basic multimedia applications (MHEG-5)".
- [9] ETS 300 777-2: "Terminal Equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 2: Use of Digital Storage Media Command and Control (DSM-CC) for basic multimedia applications".
- [10] ETS 300 777-3: "Terminal equipment (TE); End-to-end protocols for multimedia information retrieval services; Part 3: Application Programmable Interface (API) for MHEG-5".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document the definition of the standards referenced below apply. Should any ambiguity occur, definitions of the following standards apply, in decreasing order:

- DAVIC 1.3 specification [1].
- EN 300 468 [2].
- ETR 211 [3].
- ISO/IEC 13818-1 [4].

Application Programmable Interface (API): A boundary across which a software application uses facilities of programming languages to invoke software services. These facilities may include procedures or operations, shared data objects and resolution of identifiers.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
DAVIC	Digital Audio Visual Council
DSM-CC	Digital Storage Media Command and Control
DVB	Digital Video Broadcast
MHEG	Multimedia and Hypermedia information coding Experts Group
SI	Service Information
STU	Set Top Unit
VM	Virtual Machine

4 Overview

The following clause positions the API defined by this ES in the framework of the DAVIC 1.3 specification [1].

4.1 The DAVIC application interchange format

To deliver multimedia information to STUs in an interoperable way, applications shall use the MHEG-5 final form interchange format, as defined by ISO/IEC 13522-5 [6]. The ASN.1 notation and encoding, as defined by ETS 300 777-1 [8], shall be used to interchange MHEG-5 objects. This format defines the semantics and the encoding of the multimedia and hypermedia objects.

To deliver program code to STUs in an interoperable way, applications shall use the MHEG-5 `InterchangedProgram` class to encapsulate Java (note) VM code, according to the semantics and encoding defined by ISO/IEC CD 13522-6 [7]. Java VM classes are called from MHEG-5 objects using the MHEG-5 `Call` and `Fork` elementary actions.

NOTE: Java is a trade name of a product supplied by Sun Microsystems, Inc. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of the product named. Equivalent products may be used if they can be shown to lead to the same results.

The Java VM code interchange unit is a Java VM class. Java VM classes shall be encoded as defined by the Class File Format section of the Java Virtual machine specification. A Java class encapsulates data and methods that consist of sequences of instructions. The instruction set is defined by the Java Virtual machine instruction set section of the Java Virtual machine specification.

4.2 Core set of Java APIs

The following set of APIs are used by Java VM code in the DAVIC 1.3 [1] specification to express access to basic functions of the STU in an interoperable way:

- the `java.lang` package;
- the `java.util` package;

- the `iso.mheg5` package;
- the `org.davic.net.dsmcc.uu` package;
- the `org.davic.net.dvb.si` package.
- the `org.davic.mpeg` package.
- the `org.davic.mpeg.sections` package.

NOTE 1: The Java VM specification provides flexible mechanisms to call upon external functions whose interface is defined as a Java package. The DAVIC 1.3 specification only includes a minimum core set of packages required for Java VM code to be useful in a DAVIC environment. It is anticipated that additional Java packages will be standardized at a later stage.

NOTE 2: Especially, the `java.io` package, although strictly speaking not necessary to the useful performance of the VM environment, is part of the Java foundation classes. It is intended that the `java.io` package be added to the DAVIC core set of Java APIs together with an adequate specification of its semantics in a DAVIC environment.

The `java.lang` package, as defined by the Java API documentation, consists of the minimal set of Java VM classes needed to run Java VM code, supporting the following functionality: basic data types, object, mathematic operations, security, thread management, string manipulation, exception handling.

The `java.util` package, as defined by the Java API documentation, consists of Java VM classes supporting a number of utility features common to all Java VM programs.

The `iso.mheg5` package, as defined by ETS 300 777-3 [10], provides Java VM code with access to and manipulation of the MHEG-5 multimedia presentation and interaction objects, i.e. access to the dynamic attributes of MHEG-5 objects and invocation of elementary actions on MHEG-5 objects.

The `org.davic.net.dsmcc.uu` package, as defined by ETS 300 777-2 [9], enables Java VM code to use the DSM-CC U-U interface objects for network data access. This packages gives access to the DSM-CC U-U Core Consumer API as defined by ISO/IEC IS 13818-6 [5].

The `org.davic.net.dvb.si` package, as defined in this ES, enables Java VM to access information transmitted in the DAVIC Service Information (SI) stream. The DAVIC SI is based on EN 300 468 [2].

The `org.davic.mpeg` package, as defined by DAVIC [DAVIC 1.3 part 9, appendix K], provides a set of classes defining MPEG-2 PSI like objects intended for inter package references.

The `org.davic.mpeg.sections` package, as defined by DAVIC [DAVIC 1.3 part 9, appendix H], provides a set of classes allowing the filtering of MPEG-2 sections.

5 SI tables' relationships with API classes and objects

Within the SI the Network Information Table (NIT) is made of one sub-table per network (more precisely, one sub-table for the actual network -that is the network of which the transport stream containing the NIT is part of- and one sub-table per other network). A NIT sub-table corresponds to one `SINetwork` object and n `SITransportStream` objects (that is one object per transport stream carried via the network).

The Bouquet Association Table (BAT) is made of one sub-table per bouquet. A BAT sub-table corresponds to one `SIBouquet` object and n `SIService` objects (that is one `SIService` object for each service) and m `SITransportStream` objects (that is one `SITransportStream` object for each transport stream mentioned in the BAT).

The Service Description Table (SDT) is made of one sub-table per transport stream. An SDT sub-table corresponds to n `SIService` objects (that is one object per service contained in the transport stream).

The Event Information Table (EIT) is made of :

- one sub-table per service for the present event and the following event; in that case, the EIT sub-table corresponds to two `SIEvent` objects (that is one for the present event and one for the following event);
- one sub-table per service and four days period for the scheduled events; in that case, the EIT sub-table corresponds to `n` `SIEvent` objects (that is one per scheduled event).

The Time and Date Table (TDT) and the Time Offset Table (TOT) correspond to the `SITime` class.

The Running Status Table (RST) does not correspond to a class. The using application shall use the notification mechanism to be notified when the running status of an event changes.

The Stuffing Table (ST) has obviously not been modelled.

Even if it is not actually an SI table, the part of the Program Map Table (PMT) describing elementary streams (that is the second loop of the table) corresponds to `n` `PMTElementaryStream` objects (one per elementary stream).

All time values are in UTC time unless explicitly specified. The current (UTC) time can be accessed using the `java.util.Date` class.

The API allows access to the information present in the different DVB-SI tables. In DVB identifiers are used to uniquely identify an entity within the system. Each data element so identified is represented by an object in the API. The different data elements and the corresponding identifiers are:

Data element	DVB-SI table	Identified by
<code>SIBouquet</code>	BAT	<code>bouquet_id</code>
<code>SINetwork</code>	NIT	<code>network_id</code>
<code>SITransportStream</code>	NIT/BAT	<code>original_network_id</code> , <code>transport_stream_id</code>
<code>SIService</code>	SDT	<code>original_network_id</code> , <code>transport_stream_id</code> , <code>service_id</code>
<code>SIEvent</code>	EIT	<code>original_network_id</code> , <code>transport_stream_id</code> , <code>service_id</code> , <code>event_id</code>
<code>PMTElementaryStream</code>	PMT	<code>original_network_id</code> , <code>transport_stream_id</code> , <code>service_id</code> , <code>PID</code>

So an `SIEvent` object is identified by the combination of the identifiers `original_network_id`, `transport_stream_id`, `service_id`, `event_id` and not with the location in the EIT table or another mechanism.

6 Definition of `getXxx` and `retrieveXxx` methods

The API defines two different data access methods:

- `retrieve`, a retrieve method represents an optional stream access. The returned information may come from the actual transport stream.
- `get`, a get method represents a memory access, returning information that is already available.

Figure 1 explains the difference between the `get` and `retrieve` methods of the API.

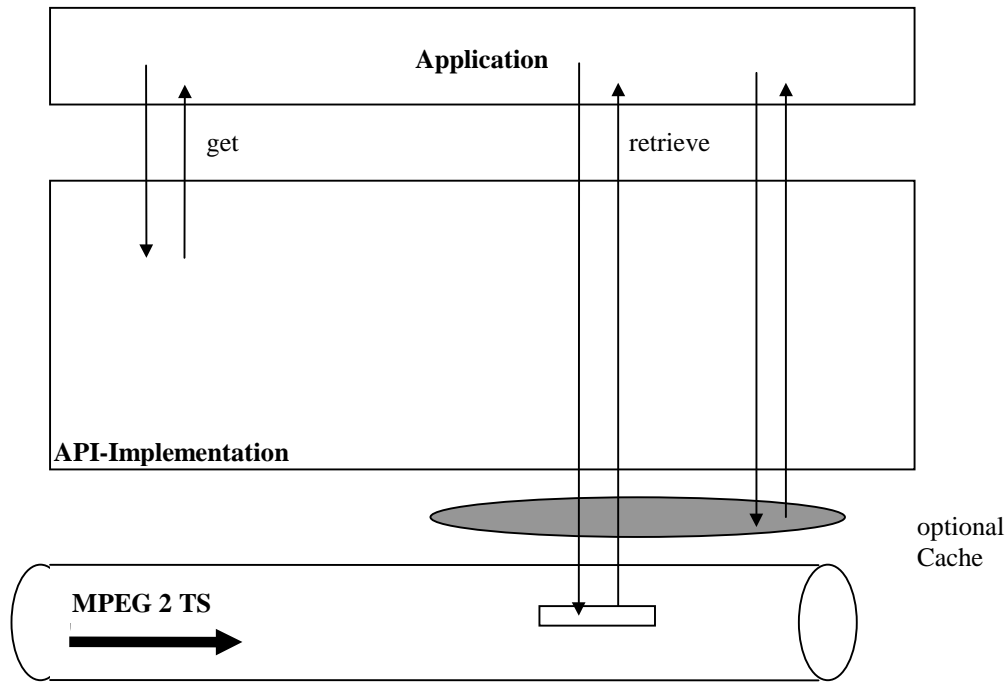


Figure 1: Semantics of get and retrieve methods

An application can use a `retrieve` method to ask the API implementation to scan the Transport Stream for specific information. The API implementation may get this information from a cache memory or by active filtering from the TS.

6.1 Different `retrieve` methods

In the API different `retrieve` methods are used that have subtle differences. To ensure interoperability the different `retrieve` methods are described in more detail. The different `retrieve` methods are:

- 1) `retrieveUpdate`, refresh the information in a `SIInformation` instance with the information in the latest version of the table-part represented by this object known to the STB.
- 2) `retrieveField` (e.g. `SIService.retrieveFreeCAMode`), and `retrieveDescriptors`, access information of the table-part represented by this object.
- 3) `retrieveSIInformation`, such as `SIDatabase.retrieveSIServices`, `SIService.retrievePresentSIEvent`, create new objects representing the latest version of a sub-table (part) known to the STB.

The `retrieve` methods represent an optional access to the transport stream. The information requested by the `retrieve` call is not always available or accessible. A set of exceptions is defined to indicate that a `retrieve` call has failed.

All exceptions indicating a failed `retrieveXxxx` call inherit from `RetrieveFailedException`. The API defines four such exceptions:

- `NotImplementedException`: this feature is not implemented on this platform.
- `LackOfResourcesException`: the resource(s) required to obtain the information are not available at this time (e.g. section filters, memory).
- `InformationSourceNoMoreAvailableException`; the requested information does not exist anymore (in cache or in the transport stream).
- `TimeoutException`: the information is not available in cache and could not be filtered within at least two times the minimum repetition rate of the DVB-SI table source as specified in ETR 211.

The first two exceptions are straightforward. The latter two have more subtle differences for the different `retrieve` method types. To explain this, the behaviour of each method type is specified separately. When the information is available in cache the behaviour is straightforward. The next paragraphs describe the behaviour of the different `retrieve` methods when an active filter action is required.

6.1.1 Method 1, `retrieveUpdate`

The data represented by the `SIInformation` instance is to be refreshed because its source table (the sub-table (part) represented by this object) may have changed. In this case an `InformationSourceNoMoreAvailableException` is thrown when the source table no longer contains the information represented by this `SIInformation` instance (e.g. the SDT no longer holds the requested service). If the source table could not be filtered within the time out period a `TimeoutException` is thrown.

6.1.2 Method 2, `retrieveField` and `retrieveDescriptors`

The application attempts to access information from within the sub-table part represented by the `SIInformation` instance. This can be a field or a descriptor. This information can be obtained from the cache or by active filtering. An `InformationSourceNoMoreAvailableException` is thrown when the version number of the source table has changed or the source table no longer contains the information represented by this `SIInformation` instance (e.g. the SDT no longer holds the requested service). If the source table could not be filtered within the time out period a `TimeoutException` is thrown.

6.1.3 Method 3, `retrieveSIInformation`

Two different methods of this type exist:

- methods returning `SIIterators`
- methods returning `SIInformation` instances (e.g. `SIService.retrieveTransportStream`, `SIEvent.retrieveSIService`, `SIService.retrievePresentSIEvent`, `SIDatabase.retrieveActualNetwork`).

If `SIIterators` are returned `InformationSourceNoMoreAvailableException` is not applicable, since a `retrieveSIInformation` call which has no matching `SIInformation` instances will return an empty `SIIterator` (note). If all data could not be obtained within the time period a `TimeoutException` is thrown.

NOTE: This also means that an application does not need to check for an `InformationSourceNoMoreAvailableException` and afterwards check if the `SIIterator` is empty.

If an `SIInformation` instance is returned an `InformationSourceNoMoreAvailableException` is thrown if the source table no longer contains the requested data (e.g. the SDT no longer holds the requested service). If the source table could not be obtained within the time period a `TimeoutException` is thrown.

6.2 Accessing descriptors

All `SIInformation` instances provide access to a sub-table part holding a descriptor loop. This descriptor loop can hold zero, one or more descriptors. Depending on the implementation some descriptors are automatically retrieved, some are not.

To optimize access to descriptors and to reduce the number of stream accesses, most `retrieve` methods allow the application to give hints at interesting descriptors. The implementation can use this information to cache these descriptors. This will probably increase performance when the application attempts to access these descriptors.

7 The API class diagram

Figure 2 shows the API class diagram using the UML (*Unified Modelling Language*) notation.

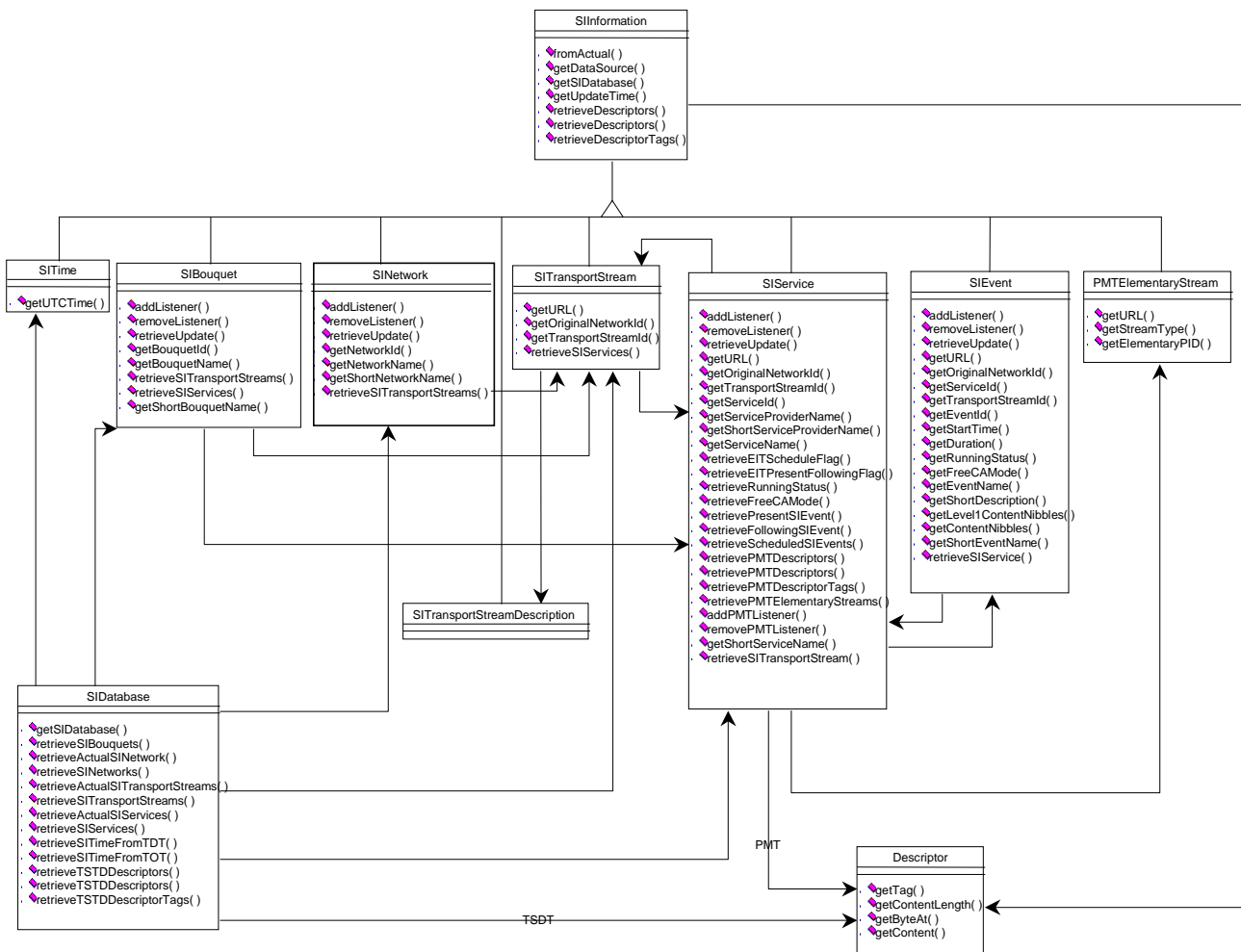


Figure 2 (continued)

The API defines the following exceptions:

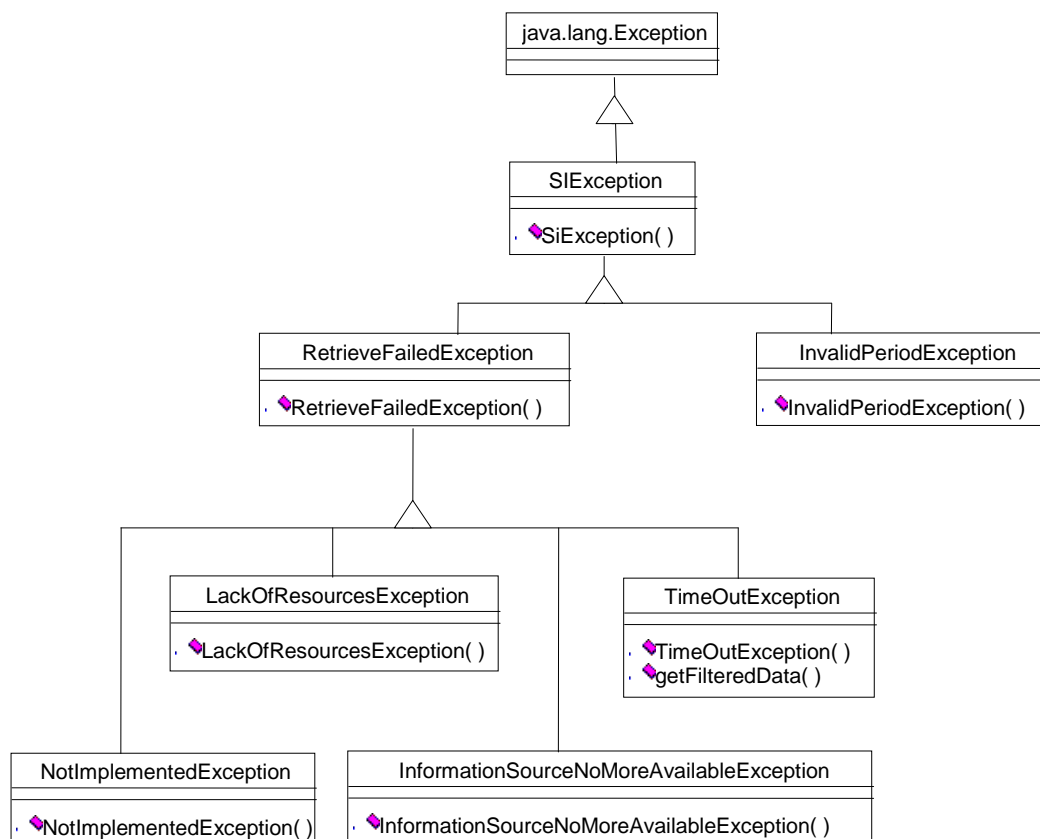


Figure 2 (continued)

Besides these classes also an event-listener interface, an iterator, and some constant defining classes are used:

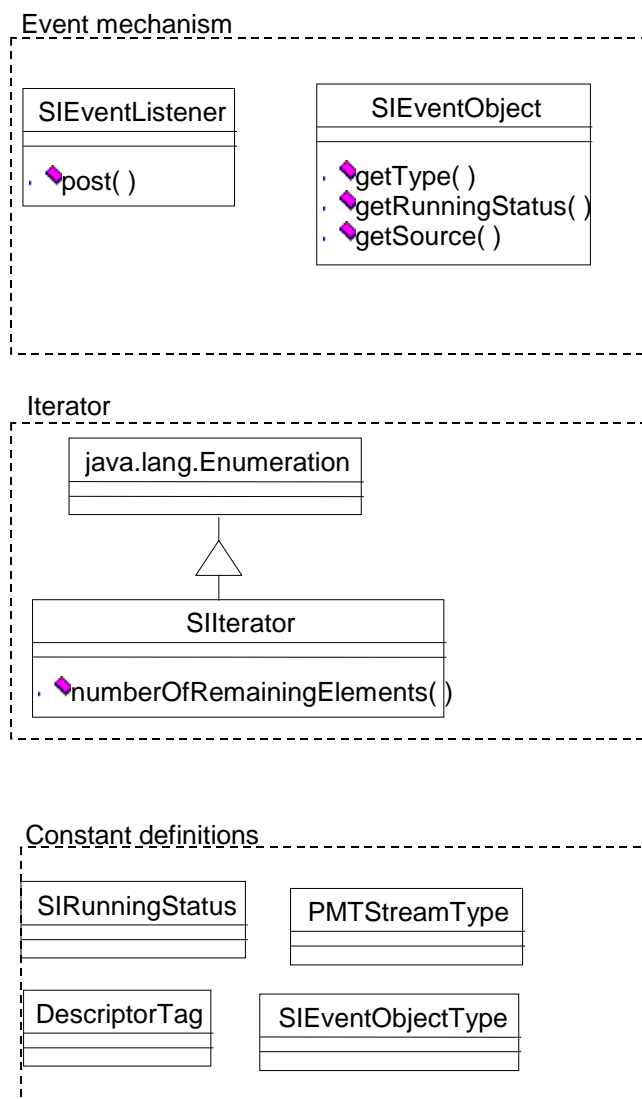


Figure 2

8 The notification mechanism

The information carried by DVB-SI sub-tables are subject to updates. Within a sub-table, the `version_number` and `current_next` indicators are used to indicate that a new version of the sub-table is available and applicable.

These changes shall be notified by the API implementation to the using application if she wants so. For that purpose, a classical event/listener mechanism has been designed. To track an API object, the application shall register (using the `addListener` method of the object to be tracked) a listener object (that is an object which implements the `SIEventListener` interface) with the object to be tracked. If an event occurs, the tracked object will then send an event object (that is an object of class `SIEventObject`) to the using application (using the `post` method of the listener object).

The notified events are:

- the tracked object does not exist any more (in that case the `getType` method of the sent event object shall return `SIEventObjectType.SOURCE_OBJECT_KILLED`);
- the running status of the information represented by the tracked object has changed (in that case the `getType` method of the sent event object shall return `SIEventObjectType.RUNNING_STATUS_CHANGED` and the `getRunningStatus` method shall return the new value of the running status); this is applicable if and only if the tracked object is an `SIEvent` or an `SIService` object; This event is also a

WARNING_ON_INFORMATION_VALIDITY event; the event also indicates the data represented by the SIEvent object may be changed.

- the information represented by the tracked object (or the information that it represents a part of) may have changed in the stream (in that case the `getType` method of the sent event object shall return `SIEventObjectType.WARNING_ON_INFORMATION_VALIDITY`).

For the last type of event the following situations are possible:

- a NIT table change has been detected : the corresponding `SINetwork` object (if tracked) will send the event; to update her related information, the application can call the `retrieveUpdate` method and the `retrieveSITransportStreams` method on the tracked object;
- a BAT sub-table change has been detected : the corresponding `SIBouquet` object (if tracked) will send the event; to update her related information, the application can call the `retrieveUpdate` method and the `retrieveSIServices` or the `retrieveSITransportStreams` method on the tracked object;
- a SDT sub-table change has been detected : all the corresponding `SIService` objects (if tracked) will send the event; to track a whole SDT sub-table the application should track only one of the corresponding `SIService` objects in order to minimize the use of filtering resources and the exchange of messages with the API implementation ; to update her related information, the application can call the `retrieveUpdate` method on the tracked object if she is only interested in a single service or recall the method she has previously used to create the set of `SIService` objects if she is interested in the whole SDT sub-table (that is the `retrieveSIServices` method of the `SITransportStream` parent object, the `retrieveSIServices` method of the `SIBouquet` parent object, or `retrieveSIServices` or the `retrieveSIActualServices` method of the `SIDatabase` parent object);
- an EIT sub-table change has been detected : all the corresponding `SIEvent` objects (if tracked) will send the event; like for a SDT sub-table, to track a whole EIT sub-table the application should track only one of the corresponding `SIEvent` objects; to update her related information, the application can call the `retrieveUpdate` method on the tracked object if she is only interested in a single event or recall the method she has previously used to create the set of `SIEvent` objects if she is interested in the whole EIT sub-table (that is the `retrieveScheduledSIEvents` method or the `retrievePresentSIEvent` and the `retrieveFollowingSIEvent` methods of the `SIService` parent object).

9 Package org.davic.net.dvb.si

9.1 Interfaces

Interface org.davic.net.dvb.si.DescriptorTag

public interface **DescriptorTag**

This interface defines constants corresponding to the most common descriptor tags.

See Also:

Descriptor

Variables

NETWORK_NAME

```
public static final short NETWORK_NAME
SERVICE_LIST
```

public static final short SERVICE_LIST
STUFFING

public static final short STUFFING
SATELLITE_DELIVERY_SYSTEM

public static final short SATELLITE_DELIVERY_SYSTEM
CABLE_DELIVERY_SYSTEM

public static final short CABLE_DELIVERY_SYSTEM
BOUQUET_NAME

public static final short BOUQUET_NAME
SERVICE

public static final short SERVICE
COUNTRY_AVAILABILITY

public static final short COUNTRY_AVAILABILITY
LINKAGE

public static final short LINKAGE
NVOD_REFERENCE

public static final short NVOD_REFERENCE
TIME_SHIFTED_SERVICE

public static final short TIME_SHIFTED_SERVICE
SHORT_EVENT

public static final short SHORT_EVENT
EXTENDED_EVENT

public static final short EXTENDED_EVENT
TIME_SHIFTED_EVENT

public static final short TIME_SHIFTED_EVENT
COMPONENT

public static final short COMPONENT
MOSAIC

public static final short MOSAIC
STREAM_IDENTIFIER

public static final short STREAM_IDENTIFIER
CA_IDENTIFIER

public static final short CA_IDENTIFIER
CONTENT

public static final short CONTENT
PARENTAL_RATING

public static final short PARENTAL_RATING
TELETEXT

public static final short TELETEXT
TELEPHONE

public static final short TELEPHONE
LOCAL_TIME_OFFSET

public static final short LOCAL_TIME_OFFSET
SUBTITLING

public static final short SUBTITLING
TERRESTRIAL_DELIVERY_SYSTEM

public static final short TERRESTRIAL_DELIVERY_SYSTEM
MULTILINGUAL_NETWORK_NAME

public static final short MULTILINGUAL_NETWORK_NAME
MULTILINGUAL_BOUQUET_NAME

public static final short MULTILINGUAL_BOUQUET_NAME
MULTILINGUAL_SERVICE_NAME

public static final short MULTILINGUAL_SERVICE_NAME
MULTILINGUAL_COMPONENT

public static final short MULTILINGUAL_COMPONENT
PRIVATE_DATA_SPECIFIER

public static final short PRIVATE_DATA_SPECIFIER
SERVICE_MOVE

public static final short SERVICE_MOVE
SHORT_SMOOTHING_BUFFER

public static final short SHORT_SMOOTHING_BUFFER
FREQUENCY_LIST

public static final short FREQUENCY_LIST
PARTIAL_TRANSPORT_STREAM

public static final short PARTIAL_TRANSPORT_STREAM
DATA_BROADCAST

public static final short DATA_BROADCAST

Interface org.davic.net.dvb.si.PMTStreamType

public interface **PMTStreamType**

This interface defines the constants corresponding to the different stream types

See Also:

PMTElementaryStream, getStreamType

Variables

MPEG1_VIDEO

```
public static final byte MPEG1_VIDEO
```

MPEG2_VIDEO

```
public static final byte MPEG2_VIDEO
```

MPEG1_AUDIO

```
public static final byte MPEG1_AUDIO
```

MPEG2_AUDIO

```
public static final byte MPEG2_AUDIO
```

Interface org.davic.net.dvb.si.SIEventListener

public interface **SIEventListener**

This interface shall be implemented by using application classes in order to listen to SI objects. When this API is implemented on a JDK 1.1 or higher platform, this object will inherit from java.util.EventListener

See Also:

SIEventObject, SIEventObjectType

Methods

post

```
public abstract void post(SIEventObject anEvent)
```

This method is called back by the listened SI object to notify the listener about an event.

Parameters:

anEvent - The notified event.

See Also:

SIEventObject, SIEventObjectType

Interface org.davic.net.dvb.si.SIEventListener

public interface **SIEventListener**

This interface shall be implemented by using application classes in order to listen to SI objects. When this API is implemented on a JDK 1.1 or higher platform, this object will inherit from java.util.EventListener

See Also:

[SIEventObject](#), [SIEventObjectType](#)

Methods

post

```
public abstract void post(SIEventObject anEvent)
```

This method is called back by the listened SI object to notify the listener about an event.

Parameters:

anEvent - The notified event.

See Also:

[SIEventObject](#), [SIEventObjectType](#)

Interface org.davic.net.dvb.si.SIInformation

public interface **SIInformation**

This interface groups the common features of SIBouquet, SINetwork, SITransportStream, SIService, SIEvent, SITime and PMTElementaryStream.

Each SIInformation instance represents a sub-table (part). Any method accessing descriptors will retrieve descriptors from the same sub-table version as the SIInformation instance. When this data is no longer available, an InformationSourceNoMoreAvailableException is thrown.

See Also:

[SIBouquet](#), [SINetwork](#), [SITransportStream](#), [SIService](#), [SIEvent](#), [SITime](#), [PMTElementaryStream](#)

Methods

retrieveDescriptors

```
public abstract SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method retrieves all descriptors in the order the descriptors are broadcast. Depending on the caching features and policy of the underlying implementation, this request can initiate an actual access to the stream.

Returns:

The Descriptor objects from the sub-table (part) of this SIIInformation instance in the order they are broadcast.

Throws: InformationSourceNoMoreAvailableException

The version number of the sub-table has changed or data represented by this SIIInformation instance is no longer present in the sub-table.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

Throws: NotImplementedException

This feature is not implemented on this platform.

See Also:

Descriptor, SIIterator

retrieveDescriptors

```
public abstract SIIterator retrieveDescriptors(short someDescriptorTags[])
throws RetrieveFailedException
```

Retrieve a set of descriptors. This method retrieves all or a set of descriptors in the order the descriptors are broadcast. Depending on the caching features and policy of the underlying implementation, this request can initiate an actual access to the stream.

If the list of tags is a subset of the one hinted to the underlying implementation (in the request which created the object on which the method is called), this is likely to increase the efficiency of the (optional) caching mechanism

Parameters:

someDescriptorTags - A list of tags for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of Descriptor objects from the sub-table (part) of this SIIInformation instance holding tags as indicated in someDescriptorTags

Throws: InformationSourceNoMoreAvailableException

The version number of the sub-table has changed or data represented by this SIIInformation instance is no longer present in the sub-table.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

Throws: NotImplementedException

This feature is not implemented on this platform.

See Also:

Descriptor, SIIterator, DescriptorTag

retrieveDescriptorTags

```
public abstract short[] retrieveDescriptorTags() throws RetrieveFailedException
```

Retrieve the tags of all descriptors that are actually broadcast for that object. The tags are returned in the same order as the descriptors are broadcast. This method returns also the tags of descriptors that were not hinted at and that are not necessarily present in the cache.

Returns:

The tags of the descriptors actually broadcast for the object (identified by their tags).

Throws: InformationSourceNoMoreAvailableException

The version number of the sub-table has changed or data represented by this SIInformation instance is no longer present in sub-table.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

Throws: NotImplementedException

This feature is not implemented on this platform.

See Also:

DescriptorTag

getSIDatabase

```
public abstract SIDatabase getSIDatabase()
```

Return the root of the hierarchy the object that implements this interface belongs to.

Returns:

The root of the hierarchy.

getUpdateTime

```
public abstract Date getUpdateTime()
```

Return the time when the information contained in the object that implements this interface was last updated.

Returns:

The date of the last update.

fromActual

```
public abstract boolean fromActual()
```

Return true when the information contained in the object that implements this interface was filtered from an 'actual' table or from a table with no 'actual/other' distinction.

Returns:

true if the information comes from an 'actual' table or from a table with no 'actual/other' distinction

getDataSource

```
public abstract TransportStream getDataSource()
```

Return the org.davic.mpeg.TransportStream object the information contained in the object that implements that interface was filtered from

Returns:

The org.davic.mpeg.TransportStream object the information was filtered from.

See Also:

TransportStream

Interface org.davic.net.dvb.si.SIiterator

```
public interface SIiterator
```

```
extends Enumeration
```

Objects implementing SIiterator interface allow to browse through a set of SI objects. In order to maintain consistency within the set of SI objects, this browsing does NOT initiate an actual access to the stream.

Methods

numberOfRemainingObjects

```
public abstract int numberOfRemainingObjects()
```

Get the number of remaining objects in the iterator.

Returns:

The number of remaining objects.

Interface org.davic.net.dvb.si.SIRunningStatus

```
public interface SIRunningStatus
```

This interface defines the constants corresponding to the running status values for services and events.

Variables

UNDEFINED

```
public static final byte UNDEFINED
```

NOT_RUNNING

```
public static final byte NOT_RUNNING
```

STARTS_IN_A_FEW_SECONDS

```
public static final byte STARTS_IN_A_FEW_SECONDS
```

PAUSING

```
public static final byte PAUSING
```

RUNNING

```
public static final byte RUNNING
```

9.2 Class Definitions

Class org.davic.net.dvb.si.Descriptor

```
java.lang.Object
|
+----org.davic.net.dvb.si.Descriptor
```

```
public class Descriptor
```

```
extends Object
```

This class represents a descriptor within a sub-table.

A descriptor consist of three fields: a tag, a contentLength and the content.

The tag uniquely identifies the descriptor type. The content length indicates the number of bytes in the content. The content consists of an array of bytes of length content length. The data represented by the content is descriptor type dependent.

See Also:

[DescriptorTag](#)

Methods

getTag

```
public short getTag()
```

Get the descriptor tag

Returns:

The descriptor tag (the most common values are defined in the DescriporTag interface)

See Also:

[DescriptorTag](#)

getContentLength

```
public short getContentLength()
```

This method returns the length of the descriptor content as coded in the length field of this descriptor.

Returns:

The length of the descriptor content.

getBytesAt

```
public byte getBytesAt(int index) throws IndexOutOfBoundsException
```

Get a particular byte within the descriptor content

Returns:

The required byte

Throws: `IndexOutOfBoundsException`

`index < 0` or `index > getContentLength()`

getContent

```
public byte[] getContent()
```

Get a copy of the content of this descriptor (everything after the length field).

Returns:

a copy of the content of the descriptor

Class `org.davic.net.dvb.si.PMTElementaryStream`

```
java.lang.Object
|
+----org.davic.net.dvb.si.PMTElementaryStream
```

```
public class PMTElementaryStream
```

```
extends Object
```

```
implements SIInformation
```

This class represents an elementary stream of a service.

For each service there is a PMT describing the elementary streams of the service. This object represents one such elementary stream. Each `PMTElementaryStream` object is uniquely defined by the combination of the identifiers `original_network_id`, `transport_stream_id`, `service_id`, `elementary_PID`.

See Also:

`SIService`, `PMTStreamType`

Methods

```
retrieveDescriptors
```

public SIIterator retrieveDescriptors() throws RetrieveFailedException

This method implements `SIInformation.retrieveDescriptors` (first prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptors

public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws RetrieveFailedException

This method implements `SIInformation.retrieveDescriptors` (second prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptorTags

public short[] retrieveDescriptorTags() throws RetrieveFailedException

This method implements `SIInformation.retrieveDescriptorTags`.

See Also:

SIInformation, retrieveDescriptorTags

getSIDatabase

public SIDatabase getSIDatabase()

This method implements `SIInformation.getSIDatabase`.

See Also:

SIInformation, getSIDatabase

getUpdateTime

public Date getUpdateTime()

This method implements `SIInformation.getUpdateTime`.

See Also:

SIInformation, getUpdateTime

fromActual

public boolean fromActual()

This method implements `SIInformation.fromActual`.

See Also:

SIInformation, fromActual

getDataSource

public TransportStream getDataSource()

This method implements `SIInformation.getDataSource`.

See Also:

SIInformation, getSource**getURL**

```
public String getURL()
```

Get the URL addressing the elementary stream.

Returns:

The URL describing this elementary stream.

getStreamType

```
public byte getStreamType()
```

Get the stream type of this elementary stream.

Returns:

The stream type (some of the possible values are defined in the PMTStreamType interface).

See Also:PMTStreamType**getElementaryPID**

```
public short getElementaryPID()
```

Get the elementary PID.

Returns:

The PID the data of elementary stream is sent on in the transport stream.

Class org.davic.net.dvb.si.SIBouquet

```
java.lang.Object
|
+----org.davic.net.dvb.si.SIBouquet
```

```
public class SIBouquet
```

```
extends Object
```

```
implements SIInformation
```

This class (together with the SITransportStream class) represents a sub-table of the Bouquet Association Table (BAT) describing a particular bouquet. Objects of this class are trackable.

Each SIBouquet object is uniquely defined by the identifier bouquet_id.

Methods

retrieveDescriptors

```
public SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method implements SIInformation.retrieveDescriptors (first prototype).

See Also:SIInformation, retrieveDescriptors

retrieveDescriptors

```
public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptors` (second prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptorTags

```
public short[] retrieveDescriptorTags() throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptorTags`.

See Also:

SIInformation, retrieveDescriptorTags

getSIDatabase

```
public SIDatabase getSIDatabase()
```

This method implements `SIInformation.getSIDatabase`.

See Also:

SIInformation, getSIDatabase

getUpdateTime

```
public Date getUpdateTime()
```

This method implements `SIInformation.getUpdateTime`.

See Also:

SIInformation, getUpdateTime

fromActual

```
public boolean fromActual()
```

This method implements `SIInformation.fromActual`.

See Also:

SIInformation, fromActual

getDataSource

```
public TransportStream getDataSource()
```

This method implements `SIInformation.getDataSource`.

See Also:

SIInformation, getDataSource

addListener

```
public void addListener(SIEventListener aListener)
```

Add a listener.

Parameters:

aListener - The object to be notified of events concerning this object.

See Also:

SIEventListener, SIEventObject

removeListener

```
public void removeListener(SIEventListener aListener)
```

Remove a listener

Parameters:

aListener - The object no longer interested in events concerning this object.

See Also:

SIEventListener, SIEventObject

retrieveUpdate

```
public void retrieveUpdate() throws RetrieveFailedException
```

Update the information contained in the object from the actual transport stream of this SIDatabase.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

Throws: NotImplementedException

This feature is not implemented on this platform.

getBouquetId

```
public int getBouquetId()
```

Get the identification.

Returns:

The bouquet identification of this bouquet.

getBouquetName

```
public String getBouquetName()
```

This method returns the name of the bouquet represented by this SIBouquet object. The name is extracted from the bouquet_name_descriptor or optionally from the multilingual_bouquet_name_descriptor. When this information is not available "" is returned. All control characters as defined in ETR 211 are ignored. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation

Returns:

The bouquet name of this bouquet.

getShortBouquetName

```
public String getShortBouquetName()
```

This method returns the short name (ETR 211) of the bouquet represented by this SIBouquet object without emphasis marks. The name is extracted from the bouquet_name_descriptor or optionally from the multilingual_bouquet_name_descriptor. When this information is not available "" is returned. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The short bouquet name of this bouquet.

retrieveSITransportStreams

```
public SIIterator retrieveSITransportStreams(short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with transport streams belonging to the bouquet.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, The application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of SITransportStream objects of the transport stream holding services of this bouquet (may be empty).

Throws: TimeoutException

The required information could not be filtered within the time out period.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

Throws: NotImplementedException

This feature is not implemented on this platform.

See Also:

SITransportStream, SIIterator, DescriptorTag

retrieveSIServices

```
public SIIterator retrieveSIServices(short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with services belonging to the bouquet.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of `SIService` objects representing the services in this bouquet (may be empty).

Throws: `TimeOutException`

The required information could not be filtered within the time out period.

Throws: `LackOfResourcesException`

The resources required to obtain the information are not available at this time.

Throws: `NotImplementedException`

This feature is not implemented on this platform.

See Also:

`SIService`, `SIIterator`, `DescriptorTag`

Class `org.davic.net.dvb.si.SIDatabase`

```
java.lang.Object
|
+----org.davic.net.dvb.si.SIDatabase
```

public class **SIDatabase**

extends `Object`

This class represents the root of the SI information hierarchy. There is one `SIDatabase` per network interface. In a system with a single network interface there is only one `SIDatabase` object.

Methods

getSIDatabase

```
public static SIDatabase[] getSIDatabase()
```

Return an array of `SIDatabase` objects (one object per network interface). In a system with one network interface, the length of this array will be one. The network interface of each `SIDatabase` is used as data source for all new data accessed by this `SIDatabase` or `SIInformation` instances obtained from it.

This is the first method to be called to access the DVB-SI API. The returned `SIDatabase` objects provide the access point to the DVB-SI information.

Returns:

An array of `SIDatabase` objects, one per network interface.

retrieveSIBouquets

```
public SIIterator retrieveSIBouquets(int bouquetId,
                                     short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with bouquets. A bouquet can be specified by its identification (when `bouquetId` is set to -1, all bouquets are retrieved)

Parameters:

bouquetId - Identification of the bouquet to be retrieved.

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored

Returns:

A set of SIBouquet objects (may be empty).

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SIBouquet, SIIterator, DescriptorTag

retrieveActualSINetwork

```
public SINetwork retrieveActualSINetwork(short someDescriptorTags[]) throws  
RetrieveFailedException
```

Retrieve information associated with the actual network. The actual network is the network carrying the transport stream currently selected by the network interface connected to this SIDatabase.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored

Returns:

The SINetwork object representing the actual network

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SINetwork, DescriptorTag

retrieveSINetworks

```
public SIIterator retrieveSINetworks(int networkId,
                                   short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with networks. A network can be specified by its identification (when networkId is set to -1, all networks are retrieved)

Parameters:

networkId - Identification of the network to be retrieved

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of SINetwork objects (may be empty).

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SINetwork, SIIterator, DescriptorTag

retrieveActualSITransportStream

```
public SITransportStream retrieveActualSITransportStream(short
someDescriptorTags[]) throws RetrieveFailedException
```

Retrieve information associated with the actual transport stream. The actual transport stream is the transport stream currently selected by the network interface connected to this SIDatabase.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The SITransportStream object representing the actual transport stream.

Throws: InformationSourceNoMoreAvailableException

The actual transport stream is no longer described in the NIT actual.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SITransportStream, DescriptorTag

retrieveSITransportStreams

```
public SIIterator retrieveSITransportStreams(int originalNetworkId,
                                           int transportStreamId,
                                           short someDescriptorTags[]) throws
```

RetrieveFailedException

Retrieve information associated with transport streams. The required transport streams can be specified by their identification (-1 means 'any' for originalNetworkId and transportStreamId). When the same transport stream exists in more than one network, a separate object will be returned for each occurrence.

Parameters:

originalNetworkId - Identification of the transport streams to be retrieved : original network identification

transportStreamId - Identification of the transport streams to be retrieved : transport stream identification

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of SITransportStream objects (may be empty).

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SITransportStream, SIIterator, DescriptorTag

retrieveActualSIServices

```
public SIIterator retrieveActualSIServices(short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with the actual services. The actual services are the services in the transport stream currently selected by the network interface connected to this SIDatabase.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The SIService objects representing the services in the actual transport stream.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SIService, SIIterator, DescriptorTag

retrieveSIServices

```
public SIIterator retrieveSIServices(int originalNetworkId,
                                     int transportStreamId,
                                     int serviceId,
                                     short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with services. The required services can be specified by their identification (-1 means 'any' for originalNetworkId, transportStreamId and serviceId)

Parameters:

originalNetworkId - Identification of the services to be retrieved : original network identification

transportStreamId - Identification of the services to be retrieved : transport stream identification

serviceId - Identification of the services to be retrieved : service identification

`someDescriptorTags` - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If `someDescriptorTags` is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of `SIService` objects (may be empty).

Throws: `TimeOutException`

The required information could not be filtered within the time out period.

Throws: `NotImplementedException`

This feature is not implemented on this platform.

Throws: `LackOfResourcesException`

The resources required to obtain the information are not available at this time.

See Also:

`SIService`, `SIIterator`, `DescriptorTag`

retrieveSITimeFromTDT

```
public SITime retrieveSITimeFromTDT() throws RetrieveFailedException
```

Retrieve information associated with time from the Time and Date Table (TDT) from the actual transport stream.

Returns:

A `SITime` object representing the TDT table in the actual transport stream.

Throws: `TimeOutException`

The required information could not be filtered within the time out period.

Throws: `NotImplementedException`

This feature is not implemented on this platform.

Throws: `LackOfResourcesException`

The resources required to obtain the information are not available at this time.

See Also:

`SITime`

retrieveSITimeFromTOT

```
public SITime retrieveSITimeFromTOT(short someDescriptorTags[]) throws RetrieveFailedException
```

Retrieve information associated with time from the Time Offset Table (TOT) from the actual transport stream. The time information will be accompanied with an offset information

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A SITime object representing the TOT table in the actual transport stream.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:SITime**retrieveSITransportStreamDescription**

```
public SITransportStreamDescription retrieveSITransportStreamDescription(short
someDescriptorTags[]) throws RetrieveFailedException
```

Retrieve the SITransportStreamDescription object representing the information of the TSDDT table in the actual transport stream of this SIDatabase object.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The SITransportStreamDescription of the actual transport stream.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SITransportStreamDescription, DescriptorTag

Class org.davic.net.dvb.si.SIEvent

```
java.lang.Object
|
+----org.davic.net.dvb.si.SIEvent
```

public class **SIEvent**

extends Object

implements SIInformation

This class represents a particular event within a service.

An object of this class represents either an event obtained from the EIT-pf or from the EIT-s. An SIEvent object filtered from the EIT-s will never be reused as the SIEvent object filtered from the EIT-pf and vice versa. Each SIEvent object is uniquely defined by the combination of the identifiers original_network_id, transport_stream_id, service_id, event_id. Objects of this type are trackable.

See Also:

SIService

Methods

retrieveDescriptors

```
public SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method implements SIInformation.retrieveDescriptors (first prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptors

```
public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws RetrieveFailedException
```

This method implements SIInformation.retrieveDescriptors (second prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptorTags

```
public short[] retrieveDescriptorTags() throws RetrieveFailedException
```

This method implements SIInformation.retrieveDescriptorTags.

See Also:

SIInformation, retrieveDescriptorTags

getSIDatabase

```
public SIDatabase getSIDatabase()
```

This method implements SIInformation.getSIDatabase.

See Also:

[SIIInformation](#), [getSIDatabase](#)

getUpdateTime

```
public Date getUpdateTime()
```

This method implements `SIIInformation.getUpdateTime`.

See Also:

[SIIInformation](#), [getUpdateTime](#)

fromActual

```
public boolean fromActual()
```

This method implements `SIIInformation.fromActual`.

See Also:

[SIIInformation](#), [fromActual](#)

getDataSource

```
public TransportStream getDataSource()
```

This method implements `SIIInformation.getDataSource`.

See Also:

[SIIInformation](#), [getDataSource](#)

addListener

```
public void addListener(SIEventListener aListener)
```

Add a listener.

Parameters:

`aListener` - The object to be notified of events concerning this object.

See Also:

[SIEventListener](#), [SIEventObject](#)

removeListener

```
public void removeListener(SIEventListener aListener)
```

Remove a listener

Parameters:

`aListener` - The object no longer interested in events concerning this object.

See Also:

[SIEventListener](#), [SIEventObject](#)

retrieveUpdate

```
public void retrieveUpdate() throws RetrieveFailedException
```

Update the information contained in the object from the actual transport stream of this

SIDatabase.

Throws: InformationSourceNoMoreAvailableException

This event is no longer present in the source EIT table (present/following or schedule).

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

Throws: NotImplementedException

This feature is not implemented on this platform.

getURL

```
public String getURL()
```

Get the URL of this event.

Returns:

The URL describing this event.

getOriginalNetworkId

```
public int getOriginalNetworkId()
```

Get the original network identification identifier.

Returns:

The original network identification.

getTransportStreamId

```
public int getTransportStreamId()
```

Get the transport stream identification identifier.

Returns:

The transport stream identification.

getServiceId

```
public int getServiceId()
```

Get the service identification identifier.

Returns:

The service identification.

getEventId

```
public int getEventId()
```

Get the event identification.

Returns:

The event identification.

getTime

```
public Date getTime()
```

Get the start time of this event in UTC time.

Returns:

The start time of this event.

getDuration

```
public long getDuration()
```

Get the duration of this event.

Returns:

The duration in milliseconds.

getRunningStatus

```
public byte getRunningStatus()
```

Get the running status of this event.

Returns:

The running status (the possible values are defined in the `SIRunningStatus` interface).

See Also:[SIRunningStatus](#)**getFreeCAMode**

```
public boolean getFreeCAMode()
```

Get the `free_CA_mode` value for this event, false indicates none of the component streams of this event are scrambled.

Returns:

The `free_CA_mode` value.

getEventName

```
public String getEventName()
```

This method returns the name of the event represented by this `SIEvent` object. The name is extracted from a `short_event_descriptor`. When this information is not available "" is returned. All control characters as defined in ETR 211 are ignored. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The event name of this event.

getShortEventName

```
public String getShortEventName()
```

This method returns the short event name (ETR 211) of the event represented by this `SIEvent` object without emphasis marks. The name is extracted from a `short_event_descriptor`. When this

information is not available "" is returned. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The short event name of this event.

getShortDescription

```
public String getShortDescription()
```

This method returns the description of the event represented by this SIEvent object. The description is extracted from a short_event_descriptor. When this information is not available, "" is returned. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation

Returns:

The short description of this event.

getLevel1ContentNibbles

```
public byte[] getLevel1ContentNibbles()
```

This method returns the level 1 content nibbles of this event. This information is extracted from the content_descriptor. If this descriptor is not present an empty array is returned (array with length 0). The return value is an array, each array element describes one content nibble. In each nibble the data occupies the four least significant bits of the returned bytes.

Returns:

All level 1 content nibbles related to the event.

getContentNibbles

```
public byte[] getContentNibbles()
```

This method returns the content nibbles related to the event. This information is extracted from the content_descriptor. If this descriptor is not present an empty array is returned (array with length 0). The return value is an array, each array element describes one content nibble. In each nibble the level 1 content nibbles occupy the four least significant bits of the returned bytes, level 2 content nibbles the four most significant bits.

Returns:

The content nibbles related to the event; level 1 content nibbles occupy the first four bits of the returned bytes, level 2 content nibbles the last four bits.

retrieveSIService

```
public SIService retrieveSIService(short someDescriptorTags[]) throws RetrieveFailedException
```

This method retrieves the SIService object representing the service the event, represented by this SIEvent, is part of.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All

non applicable tag values are ignored

Returns:

The SIService object representing the service containing the event.

Throws: InformationSourceNoMoreAvailableException

The SDT of the transport stream of this event no longer describes the required service.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SIService, DescriptorTag

Class org.davic.net.dvb.si.SIEventObject

```
java.lang.Object
|
+----org.davic.net.dvb.si.SIEventObject
```

public class **SIEventObject**

extends Object

Objects of this class are sent to listener objects of the using application to notify an event has happened. When this API is implemented on a JDK 1.1 or higher platform, this object will inherit from java.util.EventObject.

See Also:

SIEventObjectType, SIEventListener

Methods

getSource

```
public Object getSource()
```

Gets the Object that generated this Event initially occurred upon; the SIInformation instance sending the event.

Returns:

the object generating this event.

getType

```
public byte getType()
```

Get the event type of this event.

Returns:

The event type (the possible values are defined in the `SIEventObjectType` interface).

See Also:

[SIEventObjectType](#)

getRunningStatus

```
public byte getRunningStatus()
```

When this event is of the type `RUNNING_STATUS_CHANGED` this method returns the new running status of the source `SIInformation` object (the possible values are defined in the `SIRunningStatus` interface). If the event is of a different type this method returns -1.

Returns:

The running status of the source `SIInformation` object or -1.

See Also:

[SIRunningStatus](#), [getSource](#)

Class `org.davic.net.dvb.si.SINetwork`

```
java.lang.Object
|
+----org.davic.net.dvb.si.SINetwork
```

public class **SINetwork**

extends `Object`

implements [SIInformation](#)

This class (together with the `SITransportStream` class) represents a sub-table of the Network Information Table (NIT) describing a particular network. Objects of this class are trackable.

Each `SINetwork` object is uniquely defined by the identifier `network_id`.

See Also:

[SITransportStream](#)

Methods

retrieveDescriptors

```
public SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptors` (first prototype).

See Also:

[SIInformation](#), [retrieveDescriptors](#)

retrieveDescriptors

```
public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws
```

RetrieveFailedException

This method implements `SIInformation.retrieveDescriptors` (second prototype).

See Also:

[SIInformation](#), [retrieveDescriptors](#)

retrieveDescriptorTags

```
public short[] retrieveDescriptorTags() throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptorTags`.

See Also:

[SIInformation](#), [retrieveDescriptorTags](#)

getSIDatabase

```
public SIDatabase getSIDatabase()
```

This method implements `SIInformation.getSIDatabase`.

See Also:

[SIInformation](#), [getSIDatabase](#)

getUpdateTime

```
public Date getUpdateTime()
```

This method implements `SIInformation.getUpdateTime`.

See Also:

[SIInformation](#), [getUpdateTime](#)

fromActual

```
public boolean fromActual()
```

This method implements `SIInformation.fromActual`.

See Also:

[SIInformation](#), [fromActual](#)

getDataSource

```
public TransportStream getDataSource()
```

This method implements `SIInformation.getDataSource`.

See Also:

[SIInformation](#), [getDataSource](#)

addListener

```
public void addListener(SIEventListener aListener)
```

Add a listener.

Parameters:

`aListener` - The object to be notified of events concerning this object.

See Also:

[SIEventListener](#), [SIEventObject](#)

removeListener

```
public void removeListener(SIEventListener aListener)
```

Remove a listener

Parameters:

aListener - The object no longer interested in events concerning this object.

See Also:

[SIEventListener](#), [SIEventObject](#)

retrieveUpdate

```
public void retrieveUpdate() throws RetrieveFailedException
```

Update the information contained in the object from the actual transport stream of this SIDatabase.

Throws: [TimeOutException](#)

The required information could not be filtered within the time out period.

Throws: [NotImplementedException](#)

This feature is not implemented on this platform.

Throws: [LackOfResourcesException](#)

The resources required to obtain the information are not available at this time.

getNetworkId

```
public int getNetworkId()
```

Get the identification of this network.

Returns:

The network identification identifier.

getNetworkName

```
public String getNetworkName()
```

This method returns the name of the network represented by this SINetwork object. The name is extracted from the `network_name_descriptor` or optionally from the `multilingual_network_name_descriptor`. When this information is not available "" is returned. All control characters as defined in ETR 211 are ignored. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The network name of this network.

getShortNetworkName

```
public String getShortNetworkName()
```

This method returns the short name (ETR 211) of the network represented by this SINetwork object without emphasis marks. The name is extracted from the network_name_descriptor or optionally from the multilingual_network_name_descriptor. When this information is not available "" is returned. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The short network name of this network.

retrieveSITransportStreams

`public SIIterator retrieveSITransportStreams(short someDescriptorTags[]) throws RetrieveFailedException`

Retrieve information associated with transport streams carried via the network.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored

Returns:

The SITransportStream objects representing the transport streams of this network (may be empty).

Throws: TimeoutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SITransportStream, SIIterator, DescriptorTag

Class org.davic.net.dvb.si.SIService

```
java.lang.Object
|
+----org.davic.net.dvb.si.SIService
```

public class **SIService**

extends Object

implements SIInformation

This class represents a particular service carried by a transport stream. Objects of this type are trackable.

Each SIService object is uniquely identified by the combination of the following identifiers: original_network_id, transport_stream_id, service_id.

Methods

retrieveDescriptors

```
public SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method implements SIInformation.retrieveDescriptors (first prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptors

```
public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws RetrieveFailedException
```

This method implements SIInformation.retrieveDescriptors (second prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptorTags

```
public short[] retrieveDescriptorTags() throws RetrieveFailedException
```

This method implements SIInformation.retrieveDescriptorTags.

See Also:

SIInformation, retrieveDescriptorTags

getSIDatabase

```
public SIDatabase getSIDatabase()
```

This method implements SIInformation.getSIDatabase.

See Also:

SIInformation, getSIDatabase

getUpdateTime

```
public Date getUpdateTime()
```

This method implements SIInformation.getUpdateTime.

See Also:

SIInformation, getUpdateTime

fromActual

```
public boolean fromActual()
```

This method implements SIInformation.fromActual.

See Also:

SIInformation, fromActual

getDataSource

```
public TransportStream getDataSource()
```

This method implements `SIInformation.getDataSource`.

See Also:

[SIInformation](#), [getDataSource](#)

addListener

```
public void addListener(SIEventListener aListener)
```

Add a listener.

Parameters:

aListener - The object to be notified of events concerning this object.

See Also:

[SIEventListener](#), [SIEventObject](#)

removeListener

```
public void removeListener(SIEventListener aListener)
```

Remove a listener

Parameters:

aListener - The object no longer interested in events concerning this object.

See Also:

[SIEventListener](#), [SIEventObject](#)

retrieveUpdate

```
public void retrieveUpdate() throws RetrieveFailedException
```

Update the information contained in the object from the actual transport stream of this `SIDatabase`.

Throws: [InformationSourceNoMoreAvailableException](#)

The service represented by this `SIService` object no longer exists in the SDT.

Throws: [TimeOutException](#)

The required information could not be filtered within the time out period.

Throws: [NotImplementedException](#)

This feature is not implemented on this platform.

Throws: [LackOfResourcesException](#)

The resources required to obtain the information are not available at this time.

getURL

```
public String getURL()
```

Get the URL addressing the service.

Returns:

The URL of this service.

getOriginalNetworkId

```
public int getOriginalNetworkId()
```

Get the original network identification.

Returns:

The original network identification identifier.

getTransportStreamId

```
public int getTransportStreamId()
```

Get the transport stream identification.

Returns:

The transport stream identification identifier.

getServiceId

```
public int getServiceId()
```

Get the service identification.

Returns:

The service identification identifier.

getServiceName

```
public String getServiceName()
```

This method returns the name of the service represented by this SIService object. The name is extracted from the `service_descriptor` or optionally from the `multilingual_service_name_descriptor`. If this descriptor is not present "" is returned. All control characters as defined in ETR 211 are ignored. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The name of this service.

getShortServiceName

```
public String getShortServiceName()
```

This method returns the short name (ETR 211) of the service represented by this SIService object without emphasis marks. The name is extracted from the `service_descriptor` or optionally from the `multilingual_service_name_descriptor`. When this information is not available "" is returned. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The short name of this service.

getServiceProviderName

```
public String getServiceProviderName()
```


This method returns the service provider name of the service of by this SIService object. The service provider name is extracted from the `service_descriptor` or optionally from the `multilingual_service_name_descriptor`. If this descriptor is not present "" is returned. All control characters as defined in ETR 211 are ignored. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The service provider name of this service.

getShortServiceProviderName

```
public String getShortServiceProviderName()
```

This method returns the short name (ETR 211) of the service provider of by this SIService object without emphasis marks. The name is extracted from the `service_descriptor` or optionally from the `multilingual_service_name_descriptor`. When this information is not available "" is returned. For each character the DVB-SI 8 bit character code is mapped to the appropriate Unicode representation.

Returns:

The short service provider name of this service.

retrieveEITScheduleFlag

```
public boolean retrieveEITScheduleFlag() throws RetrieveFailedException
```

Retrieve the `EIT_schedule_flag` value, true indicates this services has scheduled event information.

Returns:

The `EIT_schedule_flag` value.

Throws: InformationSourceNoMoreAvailableException

The information could not be accessed because the version of the service represented by this SIService object has changed or this service no longer exists in the SDT.

Throws: TimeoutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

retrieveEITPresentFollowingFlag

```
public boolean retrieveEITPresentFollowingFlag() throws RetrieveFailedException
```

Retrieve the `EIT_present_following_flag` value, true indicates this service has present and/or following event information.

Returns:

The `EIT_present_following_flag` value.

Throws: InformationSourceNoMoreAvailableException

The information could not be accessed because the version of the service represented by this `SIService` object has changed or this service no longer exists in the SDT.

Throws: TimeoutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

retrieveRunningStatus

```
public byte retrieveRunningStatus() throws RetrieveFailedException
```

Retrieve the running status of this service.

Returns:

The running status (the possible values are defined in the `SIRunningStatus` interface)

Throws: InformationSourceNoMoreAvailableException

The information could not be accessed because the version of the service represented by this `SIService` object has changed or this service no longer exists in the SDT.

Throws: TimeoutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SIRunningStatus

retrieveFreeCAMode

```
public boolean retrieveFreeCAMode() throws RetrieveFailedException
```

Retrieve the `free_CA_mode` value of this service, false indicates none of the components of this service are scrambled.

Returns:

The `free_CA_mode` value of this service.

Throws: InformationSourceNoMoreAvailableException

The information could not be accessed because the version of the service represented by this SIService object has changed or this service no longer exists in the SDT.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

retrieveSITransportStream

```
public SITransportStream retrieveSITransportStream(short someDescriptorTags[])
throws RetrieveFailedException
```

Retrieve information associated with the transport stream carrying this service.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The SITransportStream object representing the transport stream carrying this service.

Throws: InformationSourceNoMoreAvailableException

The information could not be accessed because transport stream carrying this service is no longer present in a NIT.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SITransportStream, DescriptorTag

retrievePresentSIEvent

```
public SIEvent retrievePresentSIEvent(short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with the present event from the EIT-present/following. If the present event is not present in the EIT-present/following, an `InformationSourceNoMoreAvailableException` is thrown.

Parameters:

`someDescriptorTags` - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If `someDescriptorTags` is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The `SIEvent` object representing the present event within the service.

Throws: `InformationSourceNoMoreAvailableException`

The information could not be accessed because EIT of this service holds no present event.

Throws: `TimeOutException`

The required information could not be filtered within the time out period.

Throws: `NotImplementedException`

This feature is not implemented on this platform.

Throws: `LackOfResourcesException`

The resources required to obtain the information are not available at this time.

See Also:

`SIEvent`, `DescriptorTag`

retrieveFollowingSIEvent

```
public SIEvent retrieveFollowingSIEvent(short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with the following event from the EIT-present/following. If the following event is not present in the EIT-present/following, an `InformationSourceNoMoreAvailableException` is thrown.

Parameters:

`someDescriptorTags` - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If `someDescriptorTags` is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The `SIEvent` object representing the following event within the service.

Throws: `InformationSourceNoMoreAvailableException`

The information could not be accessed because EIT of this service holds no following event.

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SIEvent, DescriptorTag

retrieveScheduledSIEvents

```
public SIIterator retrieveScheduledSIEvents(short someDescriptorTags[],
                                           Date startTime,
                                           Date endTime) throws SIException
```

Retrieve information associated with the scheduled events within the service for a requested period from the EIT-schedule. The events are presented in the order they are present in the EIT-schedule.

Parameters:

`someDescriptorTags` - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If `someDescriptorTags` is null, the application is not interested in descriptors. All non applicable tag values are ignored.

`startTime` - The beginning of the required period in UTC time.

`endTime` - The end of the required period in UTC time.

Returns:

A set of SIEvent objects representing the scheduled events within the service for the requested period (may be empty).

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

Throws: InvalidPeriodException

When no valid period is indicated.

See Also:

SIEvent, SIiterator, DescriptorTag**retrievePMTElementaryStreams**

```
public SIiterator retrievePMTElementaryStreams(short somePMTDescriptorTags[])
throws RetrieveFailedException
```

Retrieve information associated with the elementary streams which composed the service from the Program Map Table (PMT).

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of PMTElementaryStream objects (may be empty).

Throws: TimeOutException

The required information could not be filtered within the time out period. This exception is also thrown when required the transport stream is not accessible at this moment and the information of this PMT is not available in cache.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

PMTElementaryStream, SIiterator

retrievePMTDescriptors

```
public SIiterator retrievePMTDescriptors() throws RetrieveFailedException
```

This method retrieves either all PMT descriptors.

Returns:

A set of Descriptor objects representing the descriptors in the main loop of the PMT in the order they are broadcast.

Throws: TimeOutException

The required information could not be filtered within the time out period. This exception is also thrown when required the transport stream is not accessible at this moment and the information of this PMT is not available in cache.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

Descriptor, SIIterator

retrievePMTDescriptors

```
public SIIterator retrievePMTDescriptors(short somePMTDescriptorTags[]) throws RetrieveFailedException
```

Retrieve a set of PMT descriptors from the main loop of the PMT associated with this service.

Parameters:

somePMTDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

A set of Descriptor objects with a tag value as indicated (may be empty).

Throws: TimeoutException

The required information could not be filtered within the time out period. This exception is also thrown when required the transport stream is not accessible at this moment and the information of this PMT is not available in cache.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

Descriptor, SIIterator

retrievePMTDescriptorTags

```
public short[] retrievePMTDescriptorTags() throws RetrieveFailedException
```

Retrieve the tags of all PMT descriptors that are actually broadcast for the service in the order they are broadcast.

Returns:

The list of PMT descriptors actually broadcast for the service (identified by their tags).

Throws: TimeoutException

The required information could not be filtered within the time out period. This exception is also thrown when required the transport stream is not accessible at this moment and the information of this PMT is not available in cache.

Throws: [NotImplementedException](#)

This feature is not implemented on this platform.

Throws: [LackOfResourcesException](#)

The resources required to obtain the information are not available at this time.

addPMTListener

```
public void addPMTListener(SIEventListener aListener)
```

Add a listener on the PMT associated with the Service represented by this SIService object.

Parameters:

aListener - The object interested in changes in the PMT.

See Also:

[SIEventListener](#), [SIEventObject](#)

removePMTListener

```
public void removePMTListener(SIEventListener aListener)
```

Remove a listener on the PMT associated with the Service represented by this SIService object.

Parameters:

aListener - The object no longer interested in changes in the PMT.

See Also:

[SIEventListener](#), [SIEventObject](#)

Class org.davic.net.dvb.si.SITime

```
java.lang.Object
|
+----org.davic.net.dvb.si.SITime
```

```
public class SITime
```

```
extends Object
```

```
implements SIInformation
```

This class represents the Time and Date Table (TDT) and the (optional) Time Offset Table (TOT). When this SITime object represents a TDT table, the retrieveDescriptors and retrieveDescriptorTags methods will return empty SIIterators because no descriptors can be filtered from a TDT. Note: because of the low update frequency of the TDT and TOT a filter action can only time out after 1 minute.

See Also:

[SIDatabase](#)

Methods

```
retrieveDescriptors
```



```
public SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptors` (first prototype).

See Also:

[SIInformation](#), [retrieveDescriptors](#)

retrieveDescriptors

```
public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptors` (second prototype).

See Also:

[SIInformation](#), [retrieveDescriptors](#)

retrieveDescriptorTags

```
public short[] retrieveDescriptorTags() throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptorTags`.

See Also:

[SIInformation](#), [retrieveDescriptorTags](#)

getSIDatabase

```
public SIDatabase getSIDatabase()
```

This method implements `SIInformation.getSIDatabase`.

See Also:

[SIInformation](#), [getSIDatabase](#)

getUpdateTime

```
public Date getUpdateTime()
```

This method implements `SIInformation.getUpdateTime`.

See Also:

[SIInformation](#), [getUpdateTime](#)

fromActual

```
public boolean fromActual()
```

This method implements `SIInformation.fromActual`.

See Also:

[SIInformation](#), [fromActual](#)

getDataSource

```
public TransportStream getDataSource()
```

This method implements `SIInformation.getDataSource`.

See Also:

[SIInformation](#), [getDataSource](#)

getUTCTime

```
public Date getUTCTime()
```

Get the UTC time as coded in the TDT or TOT table this object represents.

Returns:

The UTC as coded in the TDT or TOT table.

Class org.davic.net.dvb.si.SITransportStream

```
java.lang.Object
|
+----org.davic.net.dvb.si.SITransportStream
```

```
public class SITransportStream
```

```
extends Object
```

```
implements SIInformation
```

This class represents a particular transport stream belonging to a bouquet or carried via a network. A `SITransportStream` object retrieved from instances of the classes `SIDatabase`, `SINetwork`, or `SIService` represent a part of the NIT table; in this case all descriptor accessing method target the NIT. A `SITransportStream` object retrieved from an instance of `SIBouquet` represents a part of the BAT; in this case all descriptor accessing methods target the BAT.

Methods**retrieveDescriptors**

```
public SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptors` (first prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptors

```
public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptors` (second prototype).

See Also:

SIInformation, retrieveDescriptors

retrieveDescriptorTags

```
public short[] retrieveDescriptorTags() throws RetrieveFailedException
```

This method implements `SIInformation.retrieveDescriptorTags`.

See Also:

SIInformation, retrieveDescriptorTags

getSIDatabase

```
public SIDatabase getSIDatabase()
```

This method implements `SIInformation.getSIDatabase`.

See Also:

[SIInformation](#), [getSIDatabase](#)

getTime

```
public Date getTime()
```

This method implements `SIInformation.getTime`.

See Also:

[SIInformation](#), [getTime](#)

isActual

```
public boolean isActual()
```

This method implements `SIInformation.isActual`.

See Also:

[SIInformation](#), [isActual](#)

getDataSource

```
public TransportStream getDataSource()
```

This method implements `SIInformation.getDataSource`.

See Also:

[SIInformation](#), [getDataSource](#)

getURL

```
public String getURL()
```

Get the URL addressing the transport stream.

Returns:

The URL of this transport stream.

getOriginalNetworkId

```
public int getOriginalNetworkId()
```

Get the original network identification.

Returns:

The original network identification identifier.

getTransportStreamId

```
public int getTransportStreamId()
```

Get the transport stream identification.

Returns:

The transport stream identification identifier.

retrieveSIServices

```
public SIIterator retrieveSIServices(short someDescriptorTags[]) throws
RetrieveFailedException
```

Retrieve information associated with services carried via the transport stream.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The SIService objects representing the services in this transport stream (may be empty).

Throws: TimeOutException

The required information could not be filtered within the time out period.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: LackOfResourcesException

The resources required to obtain the information are not available at this time.

See Also:

SIService, SIIterator, DescriptorTag

retrieveSITransportStreamDescription

```
public SITransportStreamDescription retrieveSITransportStreamDescription(short
someDescriptorTags[]) throws RetrieveFailedException
```

Retrieve the SITransportStreamDescription object representing the information of the TSDT table of this transport stream.

Parameters:

someDescriptorTags - A list of hints for descriptors (identified by their tags) the application is interested in. If the array contains -1 as its one and only element, the application is interested in all descriptors. If someDescriptorTags is null, the application is not interested in descriptors. All non applicable tag values are ignored.

Returns:

The SITransportStreamDescription of the actual transport stream.

Throws: TimeOutException

The required information could not be filtered within the time out period. This exception is also thrown when this transport stream is not accessible at this moment and the information of this TSDT is not available in cache.

Throws: NotImplementedException

This feature is not implemented on this platform.

Throws: [LackOfResourcesException](#)

The resources required to obtain the information are not available at this time.

See Also:

[SITransportStreamDescription](#), [DescriptorTag](#)

Class org.davic.net.dvb.si.SITransportStreamDescription

```
java.lang.Object
|
+----org.davic.net.dvb.si.SITransportStreamDescription
```

public class **SITransportStreamDescription**

extends Object

implements [SIInformation](#)

This class represents the Transport Stream Description Table (TSDT).

See Also:

[SIDatabase](#), [SITransportStream](#)

Methods

retrieveDescriptors

```
public SIIterator retrieveDescriptors() throws RetrieveFailedException
```

This method implements [SIInformation.retrieveDescriptors](#) (first prototype).

See Also:

[SIInformation](#), [retrieveDescriptors](#)

retrieveDescriptors

```
public SIIterator retrieveDescriptors(short someDescriptorTags[]) throws RetrieveFailedException
```

This method implements [SIInformation.retrieveDescriptors](#) (second prototype).

See Also:

[SIInformation](#), [retrieveDescriptors](#)

retrieveDescriptorTags

```
public short[] retrieveDescriptorTags() throws RetrieveFailedException
```

This method implements [SIInformation.retrieveDescriptorTags](#).

See Also:

[SIInformation](#), [retrieveDescriptorTags](#)

getSIDatabase

```
public SIDatabase getSIDatabase()
```

This method implements `SIInformation.getSIDatabase`.

See Also:

[SIInformation](#), [getSIDatabase](#)

getTime

```
public Date getTime()
```

This method implements `SIInformation.getTime`.

See Also:

[SIInformation](#), [getTime](#)

isActual

```
public boolean isActual()
```

This method implements `SIInformation.isActual`.

See Also:

[SIInformation](#), [isActual](#)

getDataSource

```
public TransportStream getDataSource()
```

This method implements `SIInformation.getDataSource`.

See Also:

[SIInformation](#), [getDataSource](#)

9.3 Exception definitions

Class `org.davic.net.dvb.si.SIException`

```
java.lang.Object
|
+---- java.lang.Throwable
      |
      +---- java.lang.Exception
            |
            +---- org.davic.net.dvb.si.SIException
```

public abstract class **SIException**

extends `Exception`

This class is the root of the SI exceptions hierarchy.

Constructors

SIException

```
public SIException()
```

Class `org.davic.net.dvb.si.InvalidPeriodException`

```
java.lang.Object
```

```

|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----org.davic.net.dvb.si.SIException
                  |
                  +----org.davic.net.dvb.si.InvalidPeriodException

```

public class **InvalidPeriodException**

extends [SIException](#)

This exception is thrown when a specified period is invalid (for example, start time is after the end time)

Constructors

InvalidPeriodException

```
public InvalidPeriodException()
```

Class org.davic.net.dvb.si.RetrieveFailedException

```

java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----org.davic.net.dvb.si.SIException
                  |
                  +----org.davic.net.dvb.si.RetrieveFailedException

```

public abstract class **RetrieveFailedException**

extends [SIException](#)

For some reason a retrieve action has failed. This exception is the base class of all exceptions indicating a unsuccessful retrieve.

See Also:

[NotImplementedException](#), [LackOfResourcesException](#),
[InformationSourceNoMoreAvailableException](#), [TimeOutException](#)

Constructors

RetrieveFailedException

```
public RetrieveFailedException()
```

Class org.davic.net.dvb.si.NotImplementedException

```

java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----org.davic.net.dvb.si.SIException
                  |
                  +----org.davic.net.dvb.si.RetrieveFailedException

```

```

                                     +----
org.davic.net.dvb.si.NotImplementedException

```

```
public class NotImplementedException
```

```
extends RetrieveFailedException
```

This exception is thrown when the feature is not implemented on this platform.

Constructors

NotImplementedException

```
public NotImplementedException()
```

Class org.davic.net.dvb.si.LackOfResourcesException

```
java.lang.Object
```

```
|
+----java.lang.Throwable
```

```
|
+----java.lang.Exception
```

```
|
+----org.davic.net.dvb.si.SIException
```

```
|
+----org.davic.net.dvb.si.RetrieveFailedException
```

```
|
+----
```

```
org.davic.net.dvb.si.LackOfResourcesException
```

```
public class LackOfResourcesException
```

```
extends RetrieveFailedException
```

This exception is thrown when the resources required for this retrieve action were not available.

Constructors

LackOfResourcesException

```
public LackOfResourcesException()
```

Class InformationSourceNoMoreAvailableException

```
java.lang.Object
```

```
|
+----java.lang.Throwable
```

```
|
+----java.lang.Exception
```

```
|
+----org.davic.net.dvb.si.SIException
```

```
|
+----org.davic.net.dvb.si.RetrieveFailedException
```

```
|
+----
```

```
org.davic.net.dvb.si.InformationSourceNoMoreAvailableException
```

```
public class InformationSourceNoMoreAvailableException
```

```
extends RetrieveFailedException
```

This exception is thrown when table version has changed or the table source supposed to hold the requested data no longer holds this data (e.g. a SDT no longer holds the requested service).

Constructors

InformationSourceNoMoreAvailableException

```
public InformationSourceNoMoreAvailableException()
```

Class org.davic.net.dvb.si.TimeoutException

```
java.lang.Object
|
+----java.lang.Throwable
      |
      +----java.lang.Exception
            |
            +----org.davic.net.dvb.si.SIException
                  |
                  +----org.davic.net.dvb.si.RetrieveFailedException
                        |
                        +----org.davic.net.dvb.si.TimeoutException
```

```
public class TimeoutException
```

```
extends RetrieveFailedException
```

This exception is thrown when required data could not be obtained before the time out has expired. The time out duration is at least two times the minimum repetition rate for the SI table source as specified in ETR 211. The actual value is implementation dependant.

See Also:

RetrieveFailedException

Constructors

TimeoutException

```
public TimeoutException(SIIterator data)
```

Creates a new TimeoutException.

Parameters:

data - The part of the data filtered before the Timeout occurred, null indicates no data.

Methods

getFilteredData

```
public SIIterator getFilteredData()
```

This method returns the part of the data requested by the retrieve, that was filtered before the Timeout occurred.

Returns:

The filtered data or null.

10 Examples

10.1 Example 1

```

import org.davic.net.dvb.si.*;
import java.util.Enumeration;

/*****
 * Example 1
 *
 * List all service with their present event.
 * This example obtains a list of all services with the names of their present
 * event.
 *****/

public class example1{

public static void main( String[] args )
{
    SIDatabase[]    root;
    SIIterator      serviceIterator = null;
    Enumeration     list;
    SIService       service         = null;
    SIEvent         event           = null;
    String          servDesc        = null;
    java.util.Vector vecNames       = null;

    // Get the SIDatabase object
    root = SIDatabase.getSIDatabase();

    // Retrieve the list of services carried by the actual transport stream
    try
    {
        // retrieve all services
        serviceIterator = root[0].retrieveSIServices( -1, -1, -1, null);
        vecNames = new java.util.Vector();

        // Access present event of all services
        while(serviceIterator.hasMoreElements())
        {
            service = (SIService) serviceIterator.nextElement();
            servDesc = service.getServiceName();
            // Retrieve the present event
            try
            {
                event = service.retrievePresentSIEvent(null);
                // Get the name of the event
                servDesc += " : " + event.getEventName();
            }
            catch ( RetrieveFailedException e )
            {
                // no event name for this service.
            }
            vecNames.addElement( servDesc );
        }
        list = vecNames.elements();
        while ( list.hasMoreElements() )
        {
            System.out.println( (String)list.nextElement() );
        }
    }
    catch( RetrieveFailedException e )
    {

```

```

    }
}
}

```

10.2 Example 2

```

import org.davic.net.dvb.si.*;

/*****
 * Example 2
 *
 * Listener use.
 * Trace following event. This example shows an implementation of the
 * SIEventListenerInterface. This class monitors an event and sets the boolean
 * _killed when the event is no longer available.
 *****/

import org.davic.net.dvb.si.*;

public class example2
    implements SIEventListener
{
    private SIEvent _event;
    private boolean _killed;

    public example2( SIEvent event )
    {
        _event = event;
        _event.addListener( this );
    }

    public void post( SIEventObject event )
    {
        switch( event.getType() )
        {
            {
            case SIEventObjectType.RUNNING_STATUS_CHANGED:
            case SIEventObjectType.WARNING_ON_INFORMATION_VALIDITY:
                // the data in the event has changed.
                // retrieve new version
                try
                {
                    _event.retrieveUpdate();
                }
                catch ( RetrieveFailedException e )
                {
                    {}
                }
                break;
            case SIEventObjectType.SOURCE_OBJECT_KILLED:
                // event does not exist anymore.
                _killed = true;
                _event.removeListener( this );
                break;
            }
        }
    }

    // other class methods

}

```

10.3 Example 3

```

import org.davic.net.dvb.si.*;

/*****
 * Example 3
 *
 * Descriptor and field access.
 * This example shows how to access a linkage descriptor in the SDT. It searches
 * all services in the actual TS for linkage descriptors.
 *****/

import org.davic.net.dvb.si.*;

public class example3
{
static public void main( String[] args )
{
    SIDatabase[]    root;
    SIService       service = null;
    short[]         someDescriptorTags;
    SIIterator      serviceIt;
    SIIterator      eventIt;
    Descriptor      linkageDescriptor = null;

    // Get the SIDatabase object
    root = SIDatabase.getSIDatabase();
    someDescriptorTags = new short[1];
    someDescriptorTags[0] = 0x4a;
    try
    {
        // get service
        serviceIt = root[0].retrieveActualSIServices( someDescriptorTags );
        while ( serviceIt.hasMoreElements() && linkageDescriptor == null )
        {
            service = (SIService)serviceIt.nextElement();
            eventIt = service.retrieveDescriptors( someDescriptorTags );
            if ( eventIt.hasMoreElements() )
            {
                // linkage descriptor present.
                linkageDescriptor = (Descriptor)eventIt.nextElement();
            }
        }
    }
    catch ( RetrieveFailedException e )
    {
        // retrieve failed
    }
}
}

```

History

Document history		
V1.1.1	September 1998	Membership Approval Procedure MV 9844: 1998-09-01 to 1998-10-30